useR Vignette:

Reshaping Data in R

Greater Boston useR Group
April 6, 2011

by

Jeffrey Breen
jbreen@cambridge.aero

Photo from http://www.ibm-1401.info/

Outline

- Sample data from 2010 U.S. Census
  - Starts “wide”, like a spreadsheet
- reshape2 package
  - “melt” to make long
  - “*cast” to go back to wide
- Extra credit fun with “dcast”
- Further reading

Sample data: U.S. Census 2010

Read the data:

```
> pop = read.csv('http://2010.census.gov/2010census/data/pop_density.csv', skip=3)
```

Just keep the first few columns (total state populations by year):

```
> pop = pop[,1:12]
```

Clean up column names:

```
> colnames(pop)

 [1] "STATE_OR_REGION"  "X1910_POPULATION" "X1920_POPULATION" "X1930_POPULATION"
"X1940_POPULATION" "X1950_POPULATION"

 [7] "X1960_POPULATION" "X1970_POPULATION" "X1980_POPULATION" "X1990_POPULATION"
"X2000_POPULATION" "X2010_POPULATION"

> colnames(pop) = c('state', seq(1910, 2010, 10))

> colnames(pop)

 [1] "state" "1910"  "1920"  "1930"  "1940"  "1950"  "1960"  "1970"  "1980"  "1990"  "2000"
"2010"
```

Data set is “wide” like a spreadsheet

```
> head(pop,40)
                  state     1910      1920      1930      1940      1950      1960      1970      1980      1990      2000      2010
1         United States 92228531 106021568 123202660 132165129 151325798 179323175 203211926 226545805 248709873 281421906 308745538
2               Alabama  2138093   2348174   2646248   2832961   3061743   3266740   3444165   3893888   4040587   4447100   4779736
3                Alaska    64356     55036     59278     72524    128643    226167    300382    401851    550043    626932    710231
4               Arizona   204354    334162    435573    499261    749587   1302161   1770900   2718215   3665228   5130632   6392017
5              Arkansas  1574449   1752204   1854482   1949387   1909511   1786272   1923295   2286435   2350725   2673400   2915918
6            California  2377549   3426861   5677251   6907387  10586223  15717204  19953134  23667902  29760021  33871648  37253956
7              Colorado   799024    939629   1035791   1123296   1325089   1753947   2207259   2889964   3294394   4301261   5029196
8           Connecticut  1114756   1380631   1606903   1709242   2007280   2535234   3031709   3107576   3287116   3405565   3574097
9              Delaware   202322    223003    238380    266505    318085    446292    548104    594338    666168    783600    897934
10 District of Columbia   331069    437571    486869    663091    802178    763956    756510    638333    606900    572059    601723
11              Florida   752619    968470   1468211   1897414   2771305   4951560   6789443   9746324  12937926  15982378  18801310
12              Georgia  2609121   2895832   2908506   3123723   3444578   3943116   4589575   5463105   6478216   8186453   9687653
13               Hawaii   191909    255912    368336    423330    499794    632772    768561    964691   1108229   1211537   1360301
14                Idaho   325594    431866    445032    524873    588637    667191    712567    943935   1006749   1293953   1567582
15             Illinois  5638591   6485280   7630654   7897241   8712176  10081158  11113976  11426518  11430602  12419293  12830632
16              Indiana  2700876   2930390   3238503   3427796   3934224   4662498   5193669   5490224   5544159   6080485   6483802
17                 Iowa  2224771   2404021   2470939   2538268   2621073   2757537   2824376   2913808   2776755   2926324   3046355
18               Kansas  1690949   1769257   1880999   1801028   1905299   2178611   2246578   2363679   2477574   2688418   2853118
19             Kentucky  2289905   2416630   2614589   2845627   2944806   3038156   3218706   3660777   3685296   4041769   4339367
20            Louisiana  1656388   1798509   2101593   2363880   2683516   3257022   3641306   4205900   4219973   4468976   4533372
21                Maine   742371    768014    797423    847226    913774    969265    992048   1124660   1227928   1274923   1328361
22             Maryland  1295346   1449661   1631526   1821244   2343001   3100689   3922399   4216975   4781468   5296486   5773552
23        Massachusetts  3366416   3852356   4249614   4316721   4690514   5148578   5689170   5737037   6016425   6349097   6547629
24             Michigan  2810173   3668412   4842325   5256106   6371766   7823194   8875083   9262078   9295297   9938444   9883640
25            Minnesota  2075708   2387125   2563953   2792300   2982483   3413864   3804971   4075970   4375099   4919479   5303925
26          Mississippi  1797114   1790618   2009821   2183796   2178914   2178141   2216912   2520638   2573216   2844658   2967297
27             Missouri  3293335   3404055   3629367   3784664   3954653   4319813   4676501   4916686   5117073   5595211   5988927
28              Montana   376053    548889    537606    559456    591024    674767    694409    786690    799065    902195    989415
29             Nebraska  1192214   1296372   1377963   1315834   1325510   1411330   1483493   1569825   1578385   1711263   1826341
30               Nevada    81875     77407     91058    110247    160083    285278    488738    800493   1201833   1998257   2700551
31        New Hampshire   430572    443083    465293    491524    533242    606921    737681    920610   1109252   1235786   1316470
32           New Jersey  2537167   3155900   4041334   4160165   4835329   6066782   7168164   7364823   7730188   8414350   8791894
33           New Mexico   327301    360350    423317    531818    681187    951023   1016000   1302894   1515069   1819046   2059179
34             New York  9113614  10385227  12588066  13479142  14830192  16782304  18236967  17558072  17990455  18976457  19378102
35       North Carolina  2206287   2559123   3170276   3571623   4061929   4556155   5082059   5881766   6628637   8049313   9535483
36         North Dakota   577056    646872    680845    641935    619636    632446    617761    652717    638800    642200    672591
37                 Ohio  4767121   5759394   6646697   6907612   7946627   9706397  10652017  10797630  10847115  11353140  11536504
38             Oklahoma  1657155   2028283   2396040   2336434   2233351   2328284   2559229   3025290   3145585   3450654   3751351
39               Oregon   672765    783389    953786   1089684   1521341   1768687   2091385   2633105   2842321   3421399   3831074
40         Pennsylvania  7665111   8720017   9631350   9900180  10498012  11319366  11793909  11863895  11881643  12281054  12702379
```

...and wide can be convenient

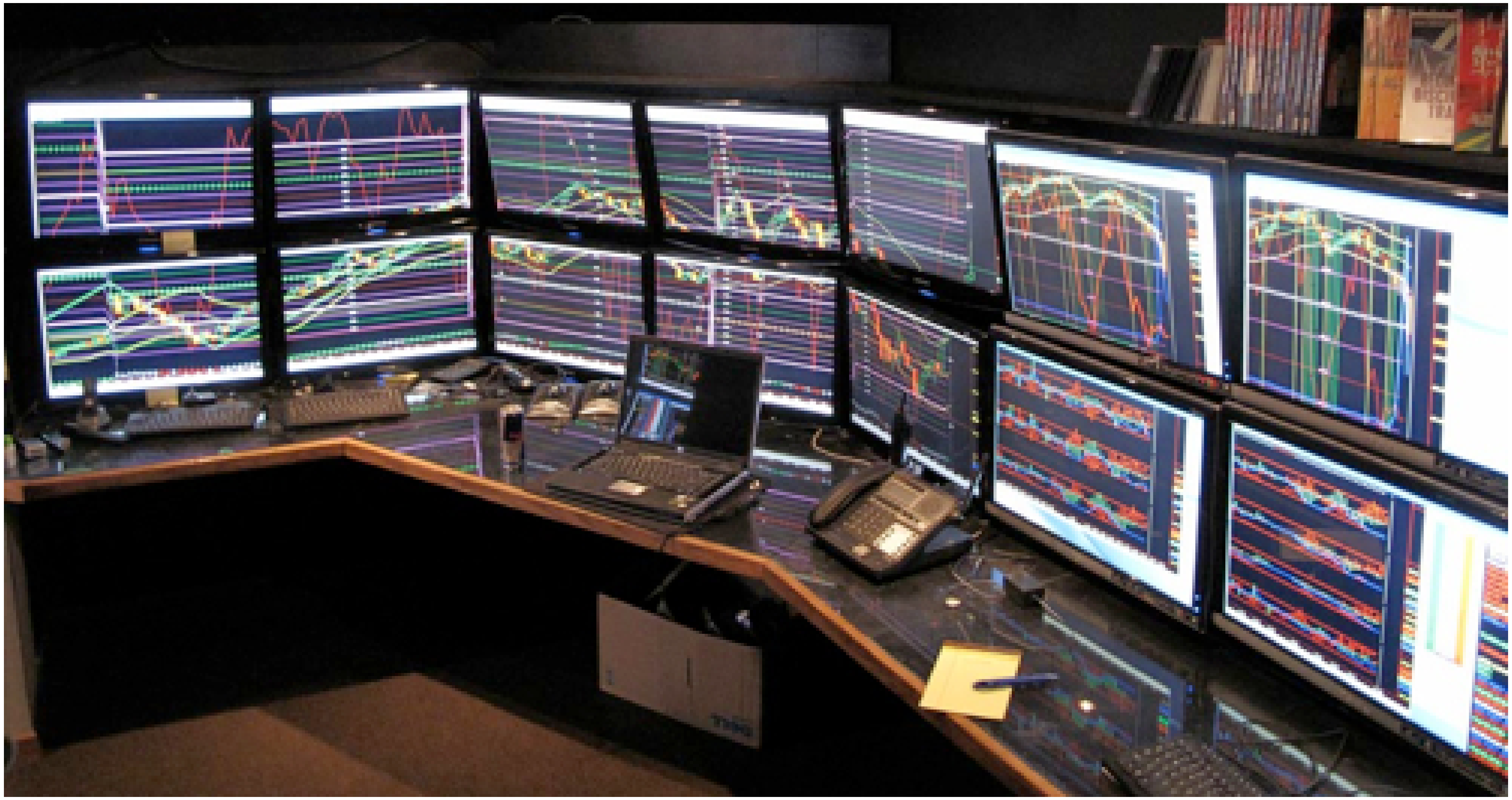
- Wide format can be useful – just like spreadsheets
- Easy to read, comprehend
- Can sort whole data set by one column:

```
> library(doBy)
> top = orderBy(~-2010, pop)

> top = subset(top, state!='United States')
> top = head(top, 10)
> top$state = factor(top$state)

> head(top)
          state    1910     1920     1930     1940     1950     1960     1970     1980     1990     2000     2010
6    California 2377549  3426861  5677251  6907387 10586223 15717204 19953134 23667902 29760021 33871648 37253956
45        Texas 3896542  4663228  5824715  6414824  7711194  9579677 11196730 14229191 16986510 20851820 25145561
34     New York 9113614 10385227 12588066 13479142 14830192 16782304 18236967 17558072 17990455 18976457 19378102
11      Florida  752619   968470  1468211  1897414  2771305  4951560  6789443  9746324 12937926 15982378 18801310
15     Illinois 5638591  6485280  7630654  7897241  8712176 10081158 11113976 11426518 11430602 12419293 12830632
40 Pennsylvania 7665111  8720017  9631350  9900180 10498012 11319366 11793909 11863895 11881643 12281054 12702379
```

- But how would you plot population vs. year?

...or too much of a good thing

I'm sure this seemed like a good idea at the time



reshape2 package

- Hadley Wickham's “reboot of the reshape package”
  - Like “plyr”, naming convention denotes output data type: acast() → arrays, dcast() → data.frames
    - Beware conflict between reshape::melt() and reshape2::melt()
- Announced September 2010 on [R-pkgs]:
  - http://r.789695.n4.nabble.com/R-pkgs-reshape2-a-reboot
- Similar functions in Base R
  - utils::stack(), utils::unstack(), stats::reshape()

melt() all those columns away

melt() treats column names as a variable as it collapses data into long format:

```
> mtop = melt(top, id.vars='state', variable.name='year', value.name='population')
```

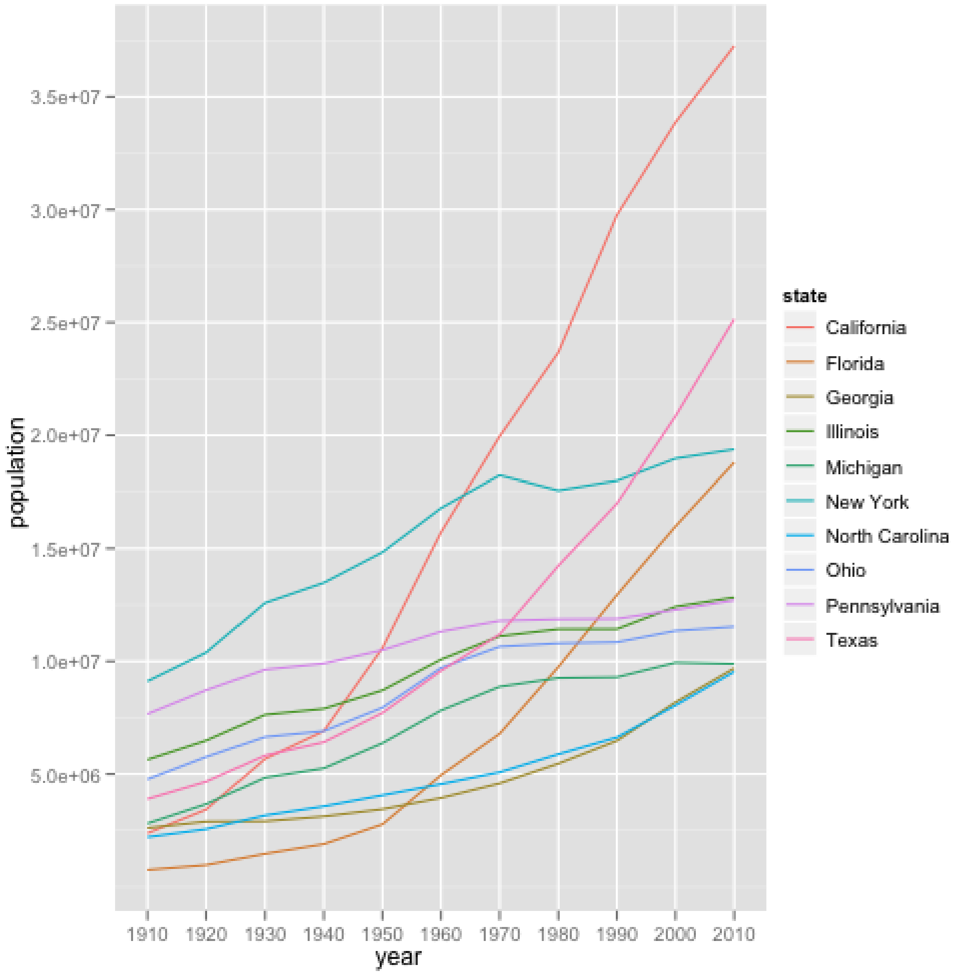
```
> head(mtop)
         state year population
1   California 1910    2377549
2        Texas 1910    3896542
3     New York 1910    9113614
4      Florida 1910     752619
5     Illinois 1910    5638591
6 Pennsylvania 1910    7665111
> tail(mtop)
             state year population
105       Illinois 2010   12830632
106   Pennsylvania 2010   12702379
107           Ohio 2010   11536504
108       Michigan 2010    9883640
109        Georgia 2010    9687653
110 North Carolina 2010    9535483
```

Long, “molten” format (may be) easier for analysis, plotting, database storage, etc.

Obligatory graph

Molten form certainly makes graphing easier in ggplot2:

```
library(ggplot2)

ggplot(data=mtop, aes(group=state)) +
geom_line(aes(x=year, y=population,
color=state))
```



“cast” functions put data back into wide form

- Sometimes you really can put the toothpaste back into the tube
  - Use acast() to produce arrays/matrices, dcast() for data.frames
  - Accepts formula notation

```
> dcast(mtop, state~year, value_var='population')
            state    1910     1920     1930     1940     1950     1960     1970     1980     1990     2000     2010
1      California 2377549  3426861  5677251  6907387 10586223 15717204 19953134 23667902 29760021 33871648 37253956
2         Florida  752619   968470  1468211  1897414  2771305  4951560  6789443  9746324 12937926 15982378 18801310
3         Georgia 2609121  2895832  2908506  3123723  3444578  3943116  4589575  5463105  6478216  8186453  9687653
4        Illinois 5638591  6485280  7630654  7897241  8712176 10081158 11113976 11426518 11430602 12419293 12830632
5        Michigan 2810173  3668412  4842325  5256106  6371766  7823194  8875083  9262078  9295297  9938444  9883640
6        New York 9113614 10385227 12588066 13479142 14830192 16782304 18236967 17558072 17990455 18976457 19378102
7  North Carolina 2206287  2559123  3170276  3571623  4061929  4556155  5082059  5881766  6628637  8049313  9535483
8            Ohio 4767121  5759394  6646697  6907612  7946627  9706397 10652017 10797630 10847115 11353140 11536504
9    Pennsylvania 7665111  8720017  9631350  9900180 10498012 11319366 11793909 11863895 11881643 12281054 12702379
10          Texas 3896542  4663228  5824715  6414824  7711194  9579677 11196730 14229191 16986510 20851820 25145561
```

Extra credit: *cast functions for BI

- Disclaimer on http://had.co.nz/reshape/ warns not a “fully fledged OLAP solution”
  - But *cast() can replace table() for computing frequency/contingency tables and crosstabs
  - Formula notation allows you to pick out specific columns so wide data can look molten
- Here's some (fake) consumer survey data:

```
> head(survey)
           ResponseID    sex         age favorite.airline
1 R_51JpA6GecA0SRcU   Female 36-49 years   Virgin America
2 R_0N77P8pZyPnjctm     Male 50-65 years        Southwest
3 R_eFoGuGRSuzqnHgM     Male 50-65 years        Southwest
4 R_9KXgybRXPiDG3LS     Male 36-49 years      Continental
5 R_cCH0fRZmc0zwGkk     Male 50-65 years  Delta/Northwest
6 R_ba8ujmCV5OnBNxW     Male 36-49 years        Southwest
[...]
```

Crosstabs with dcast()

```
> dcast(survey, favorite.airline~sex, value_var='favorite.airline', fun.aggregate=length)
   favorite.airline Male Female
1          Air Tran    7      2
2   Alaska Airlines   17      3
3         Allegiant    0      1
4          American   20      9
5       Continental   31      8
6   Delta/Northwest   36      6
7  Frontier/Midwest    4      3
8           JetBlue   41     29
9         Southwest  100     40
10           Spirit    1      0
11           United   23     10
12       US Airways    2      4
13   Virgin America   30     18

> dcast(survey, favorite.airline~age, value_var='favorite.airline', fun.aggregate=length)
   favorite.airline 18-24 years 25-35 years 36-49 years 50-65 years 66+ years
1          Air Tran           0           1           3           4         1
2   Alaska Airlines           0           0           5          11         4
3         Allegiant           0           0           1           0         0
4          American           1           4           7          14         3
5       Continental           0           3          15          17         4
6   Delta/Northwest           0           6          16          20         0
7  Frontier/Midwest           0           1           3           3         0
8           JetBlue           2          12          19          35         2
9         Southwest           0          20          44          66        10
10           Spirit           0           0           0           1         0
11           United           2           4          13          14         0
12       US Airways           0           0           3           3         0
13   Virgin America           0           8          23          13         4
```

Further reading

- reshape2 package on CRAN
  - http://cran.r-project.org/web/packages/reshape2/
- Hadley's github (bleeding edge)
  - https://github.com/hadley/reshape
- Decision Stats: “Using Reshape2 for transposing datasets in R”
  - http://decisionstats.com/2010/11/06/using-reshape2-for-transposir
- Recology: “Good riddance to Excel pivot tables”
  - http://r-ecology.blogspot.com/2011/01/good-riddance-to-excel-pivo
- Stack Overflow discussions: “[r] reshape2”
  - http://stackoverflow.com/search?tab=votes&q=[r]%20reshape2